

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

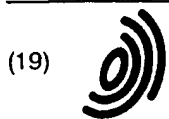
Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 1 056 022 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.11.2000 Bulletin 2000/48

(51) Int. Cl.⁷: G06F 17/30, G06F 17/24

(21) Application number: 00110966.9

(22) Date of filing: 26.05.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 27.05.1999 JP 14844299

(71) Applicants:
• Rlp Japan Co., Ltd.
Oota-Ku, Tokyo 144-0051 (JP)

• Yukino, Kensaku
Shinjuku-ku, Tokyo 162-0801 (JP)

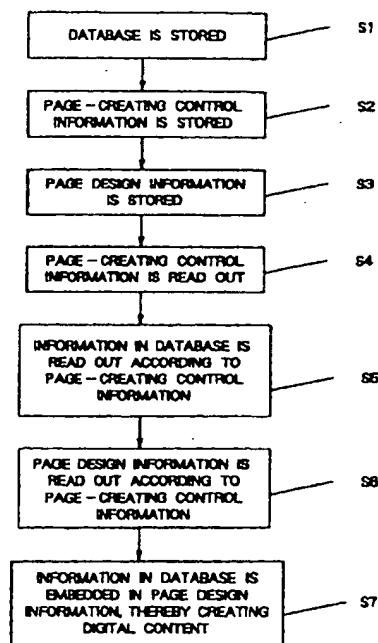
(72) Inventor: Yukino, Kensaku
Tokyo 162-0801 (JP)

(74) Representative:
Casalonga, Axel et al
BUREAU D.A. CASALONGA - JOSSE
Morassistrasse 8
80469 München (DE)

(54) Automatic digital content-organizing system and record medium having automatically organized digital content recorded therein

(57) An automatic digital content-organizing system and a record medium having automatically organized digital content recorded therein, which is designed to allow a user and a Web designer to easily change design of a created HTML page without the aid of a programmer. The automatic digital content-organizing system according to the present invention creates HTML pages by the steps of: indicating to a program types of the HTML pages and a method for creating the HTML pages, all of which HTML pages form a site; defining designs of the created HTML pages in combination with such indicated types and method; regularly extracting data such as commodities from a single or multiple databases; and, permitting the extracted data to be embedded in the HTML pages based on templates that include such defined designs.

FIG. 2



EP 1 056 022 A2

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] This invention relates to a digital content-organizing system. More particularly, it relates to an automatic digital content-organizing system and a record medium having automatically organized digital content recorded therein, such as an on-line system (e.g., WWW or World Wide Web on the Internet and personal computer-to-personal computer communication), an electronic catalog, and a CD (compact disk)-ROM.

2. Description of the Related Art

[0002] The widespread use of personal computers and the Internet causes commerce via networks what is called on-line shopping to be becoming much more universal at an increasing rate. In such on-line shopping, enumerated commodities are displayed on a homepage. In addition, commodity catalogs are provided as electronic data to purchasers.

[0003] Digital content such as the homepage and electronic catalog for use in such on-line shopping has heretofore been made up by either a manual organizing process or a dynamic organizing process based on a database. In the manual organizing process, individual pages are manually prepared, and further a menu page linked to such individual pages is manually drawn up. In this way, the homepage is organized. In the dynamic organizing process in which the homepage is created on the basis of the database, a variety of proposals such as Active Server Page (hereinafter called "ASP") provided by the Microsoft Inc. as well as CGI (Common Gateway Interface) has been made and then utilized.

[0004] However, a first problem incurred with the dynamic organizing process is that design of a created HTML page is difficult to change. In addition, an HTML page having a complicated design is difficult to create.

[0005] For CGI, in some cases, CGI itself is written in script language such as "perl." In other cases, CGI is a dedicated execute form program that is described in C-language and is then compiled. As a result, design of a created HTML file is embedded in a program by a system developer, and it is nearly impossible for a user to change the design. Further, it is laborious for the system developer to change the design, and a relatively upper amount of development cost is required to pay when the user passes an order to the system developer to change the design.

[0006] For ASP, the user is usually disallowed to change the design because database retrieval and output specification are described in the HTML file. As a result, the HTML page produced by CGI and ASP is often poor in design, and thus less appeals to the viewer. In addition, creation of an elaborate design results in high cost because man-day for development is increased.

[0007] A second problem with the dynamic organizing process is that a site page structure is fixed, and is difficult for the user to change. For example, assume that there is provided a Web site page structure created by a plurality of HTML pages such as a top page of "index.html" and subsequent pages that include a retrieval-indicating page, a retrieval result-displaying page, a descriptive page as to how an order is placed, and a shop guide page. Then, a portion of the above page structure, which is created by CGI, is difficult to change.

[0008] In view of the above, an object of the present invention is to provide an automatic digital content-organizing system and a record medium having automatically organized digital content recorded therein, which is designed to allow a user and a Web designer to easily change design of a created HTML page without the aid of a programmer.

[0009] A further object of the present invention is to provide an automatic digital content-organizing system and a record medium having automatically organized digital content recorded therein, which is designed to allow a user and a Web designer to easily change a structure and design of a HTML page to be created by referenced data in a database being embedded in the HTML page.

SUMMARY OF THE INVENTION

[0010] In order to achieve the above objects, an automatic digital content-organizing system according to the present invention creates HTML pages by the steps of: indicating to a program types of the HTML pages and a method for creating the HTML pages, all of which HTML pages form a site; defining designs of the created HTML pages in combination with such indicated types and method; regularly extracting data such as commodities from a single or multiple databases; and, permitting the extracted data to be embedded in the HTML pages based on templates that include such defined designs.

[0011] More specifically, an aspect of the present invention provides an automatic digital content-organizing system comprising:

a means for preparing digital content including a character and an image portion by means of either HTML or any other protocol-based digital content description language for purposes of information processing;

a database information storage means for storing a single or multiple databases, the database being a source of information for creating the digital content;

a management means for administering retrieval of data from the database information storage means and input-output of the data from the database information storage means;

a page-creating control information storage means for storing page-creating control information, the page-creating control information containing a page structure and a method for creating the page structure, the page structure consisting of a plurality of kinds of pages;

a page design information storage means for storing page design information in a template that contains two tags, in which one of the tags specifies respective designs of the page structure and respective positions at which fields of the database are displayed, while another tag is a link failure prevention-indicating tag that functions to avoid including a description of a link to image information and/or any other page when the image information and/or other page is absent;

a means for reading out the page-creating control information from the page-creating control information storage means;

a means for reading out information in the database from the database information storage means according to the page-creating control information, in which the information in the database is read out in a predetermined format under control of the management means;

a means for reading out optimum page design information from the page design information storage means according to the page-creating control information; and,

a program-storing means for storing a program, the program permitting such read information in the database to be embedded in the page design information, the program thereby producing pages that form the digital content.

[0012] A preferred embodiment of the present invention provides an automatic digital content-organizing system as described above, further comprising:

the page design information storage means for storing a template that contains a substitution-specifying tag, the substitution-specifying tag replacing a predetermined field of the database with a predetermined description, and then displaying such a replaced description when the database has a predetermined value in the predetermined field; and,

the program-storing means for storing a program that executes such substitutional specification.

[0013] A further preferred embodiment of the present invention provides an automatic digital content-organizing system as previously described, further comprising:

the page-creating control information storage means for storing the page-creating control information, to which condition output-indicating information is added, the condition output-indicating information determining on the basis of a value in a particular field of the database whether a particular page is created;

the page design information storage means for storing the page design information, to which a condition output-indicating tag is added, the condition output-indicating tag indicating depending upon a value in a particular field of the database whether a predetermined character string including a link to any other page is output; and,

the program-storing means for storing a program that executes such condition output indication.

[0014] Another aspect of the present invention provides a record medium having digital content recorded therein, which digital content is automatically organized by the above-described automatic digital content-organizing system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] This and other object, features, and advantages of the present invention will become more apparent upon a reading of the following detailed description and drawings, in which:

Fig. 1 is a block diagram, illustrating a system for automatically organizing an on-line shopping Web site according to the present invention;

Fig. 2 is a flowchart, showing one embodiment of an automatic digital content-organizing system according to the present invention;

Fig. 3 is an illustration, showing a site structure created according to embodiments given in this specification;

Fig. 4 is an illustration, showing definitions in fields of a commodity database;

Fig. 5 is an illustration, showing a first page of definitions in fields of a shop database;

Fig. 6 is an illustration, showing a second page of definitions in fields of the shop database;

Fig. 7 consisting of Figs. 7A to 7D is an illustration, showing one embodiment of a first page of a page-creating control file drawn up as page-creating control information;

Fig. 8 consisting of Figs. 8A to 8D is an illustration, showing a second page of the page-creating control file;

Fig. 9 is an illustration, showing one embodiment of a first page of a template of a "top page";

Fig. 10 is an illustration, showing a second page of the template;

Fig. 11 is an illustration, showing a third page of the template;

Fig. 12 is an illustration, showing one embodiment of a first page of a template of a "commodity page";

Fig. 13 is an illustration, showing a second page of the template;

Fig. 14 is an illustration, showing a third page of the template;

Fig. 15 is an illustration, showing a fourth page of the template;

Fig. 16 is an illustration, showing a fifth page of the template;

Fig. 17 is an illustration, showing a "top page" using a browser, which top page is created by the automatic digital content-organizing system according to the present invention; and,

Fig. 18 is an illustration, showing a "commodity page" by means of the browser, which commodity page is generated by the automatic digital content-organizing system according to the present invention.

[0016] Embodiments of the present invention will be described in detail with appropriate reference to the drawings.

Fig. 1 is a block diagram, illustrating a system for automatically organizing an on-line shopping Web site which includes two main databases and two subsidiary databases. The main databases include a shop database and a commodity database. The shop database stores information on an on-line shopping store. The subsidiary databases include upper and lower classification tables, which are referenced from the commodity database.

[0017] The above system includes a CPU 1, a boot ROM 2, a RAM 3, a disk 4, a display (CRT) 5, and a keyboard (KB) 6. The boot ROM 2 is read when the system is activated. Further, when the system is activated, then an OS (operating system) 15 is read from the disk 4, while the RAM 3 is used as a working area for various programs. The display 5 and the keyboard 6 are responsible for a man-machine interface.

[0018] The disk 4 stores data or programs that follow: a shop database (shopinfo.DB) 7; a commodity database (catalog.DB) 8; a upper classification table 9; a lower classification table 10; a database management system (DBMS) 11; a data file of page-creating control information 12; a template file as page design information 13; and, a page-creating program 14. The upper and lower classification tables 9, 10 are referenced from the commodity database 8. The database management system (DBMS) 11 is actuated when data are fed into and out of the database. The page-creating program 14 functions to create pages.

[0019] Fig. 2 is a flowchart, showing one embodiment of an automatic digital content-organizing system according to the present invention. At step S1, a single or multiple databases as a source of information for creating digital content is or are stored. The database is either saved in advance by a user or reproduced from any other storage means to a database information storage means that constitutes the automatic digital content-organizing system. At step S2, the page-creating control information containing a page structure and a method for creating the page structure is stored. The page structure consists of a plurality of kinds of pages such as a top page, annexed pages, commodity menu pages, and a list page enumerating the commodity menu pages. At step S3, the page design information in a template that contains tags is stored. One of the tags specifies respective designs of the above-described page structure and respective positions at which fields of the database are displayed. The other tag is a link failure prevention-indicating tag as a preventive against a disordered display on a screen which otherwise would be attributable to a link error in the absence of image information and linked pages, as detailed afterwards. At step S4, the page-creating control information is read out from a page-creating control information storage means. At step S5, information in the database is read out from the database information storage means according to the page-creating control information that is stored at step S2. In this case, the information in the database is read out in a predetermined format under the control of any management means such as the database management system. At step S6, optimum page design information is read out from a page design information storage means according to the page-creating control information that is stored at step S2. At step S7, the information in the database, which has been read out at step S5, is embedded in the page design information, thereby producing pages that form digital content.

[0020] A preferred embodiment of the present invention is that a template containing a substitution-specifying tag is stored at step S3. The substitution-specifying tag replaces a predetermined field of the database with a predetermined description, and then displays such a replaced description when the database has a predetermined value in the above-mentioned predetermined field. A further preferred embodiment is that such substitutional specification is executed at step S7.

[0021] A yet further preferred embodiment is that the page-creating control information having condition output-indicating information added thereto is stored at step S2. The condition output-indicating information determines, on the

basis of a value in a particular field of the database, whether a particular page is created. Another desirable embodiment of the present invention is that the page design information having a condition output-indicating tag added thereto is stored at step S3. The condition output-indicating tag indicates, depending upon a value in a particular field of the database, whether a predetermined character string including a link to any other page is output. A still further desirable embodiment is that such condition output indication is executed at step S7.

[0022] In the above embodiment, page-creating control and page design are indicated and then executed by a program at step S7. A page-creating control means for providing such page-creating control indicates the following for each kind of a page to be created: a type of the page; a title of the page; a created HTML filename; conditions to select a record of a commodity database to be output; conditions to recognize whether the page is updated; a template filename to be used; the number of output (blocking factor) in which a plurality of records of the commodity database are output to a page in list pages; and, field names of the commodity database to be output when the above records of the commodity database are output to the page. The page-creating control means corresponds to a programmatic description for CGI and ASP.

[0023] A page-designing means for drawing the above-mentioned page design includes a template file and a memory or any other storage medium for storing the template file. The template file is an HTML file in which variable tags are contained and embedded for each kind of a page to be created. The page-designing means corresponds to a descriptive portion of an HTML file for CGI and ASP.

[0024] A template stored in the page design information storage means contains a description characteristic of the present invention. More specifically, such a characteristic description include the link failure prevention-indicating tag and a character string substitution-indicating tag. The former tag prevents a display on a screen from being disordered when an image file is absent. In addition, in a menu page in which commodity names are enumerated from a plurality of records of the commodity database, the link failure prevention-indicating tag inhibits a description of a link from one commodity page to another from being created when the number of retrieval is fewer than the blocking factor. Meanwhile, the character string substitution-indicating tag indicates replacement of data in the database, and such indication only written to a template provides a display comparable to that is acquired when a partitioned table is look up. Further, the character string substitution-indicating tag makes a change in design. Yet further, the template stored in the page design information storage means contains a condition output-indicating tag. This tag indicates whether a particular page is created, depending upon a value in a predetermined field of a shop database. Thus, the above-described template creates a perfect HTML file, and a site structure is controlled. Further details on the above-mentioned three types of tags are discussed afterwards.

[0025] Fig. 3 illustrates a site structure that is generated according to embodiments given in this specification. A top page is an index.htm page. The top page is shown branched into links in a tree fashion. A list of the upper classification table is displayed in the top page.

[0026] In Fig. 3, the annexed pages ranging from a site guide to a home visit sales law are limited to displaying information on a shop database. For membership system-related and order-related pages among the annexed pages, creation of such pages is controlled by the condition output-indicating tag that is placed in an under-mentioned field of "processing condition" of a page-creating control file.

[0027] As illustrated by a tree structure in Fig. 3, the upper classification page displays a list of lower classification pages, and thus may be called a lower classification list page. In the upper classification page, a list of lower classifications is displayed, which are branched from a commodity that belongs to a certain upper classification. Meanwhile, as illustrated by the tree structure in Fig. 3, the lower classification page displays a list of commodity pages, and thus may be called a commodity list page. The lower classification page is a commodity-enumerated page in which a predetermined combination of upper and lower classifications are provided.

[0028] Fig. 4 is an illustration, showing definitions in fields of a commodity database. A feature of the definitions is that one field of "record-updating time & date" and another field of "local page-updating time & date" are provided to minimize commodity pages in order to avoid creating unnecessary pages. A further feature of the definitions is that a field of "custom template" is provided. The custom template is used when a user or a Web designer wants to change a page design for each record of the commodity database.

[0029] Figs. 5 and 6 are illustrations, showing definitions in fields of a shop database. In the definitions, important fields are "processing or unprocessing of membership registration" and "presence or absence of order page." Respective values in these fields are used for undermentioned condition output indication.

[0030] Some shopping sites are provided for membership registration, but others are not. Further, aside from the shopping site, there may be a desire to establish a catalog site without the ability to accept orders. In such cases, a predetermined value (01) is entered into the above fields of the shop database, thereby creating membership registration-related and order-related pages.

[0031] In the commodity database, a field of "page update-reserving date & time" is provided to permit data on a new commodity to be opened to the public on scheduled date and time, which data are saved in advance. In addition, the field of "page update-reserving date & time" is provided to permit seasonal commodities to be changed at one time

on a specified date.

[0032] In the commodity database, a field of "option" is used to create a new commodity page. In this case, a commodity page is generated for each commodity. In addition, in such a page, detailed information on a commodity including an image of the commodity is displayed. Regarding the commodity image, three or fewer images can be registered, and are then usable within the range of such a quantity, depending upon intended use of each commodity. The under-mentioned link failure prevention-indicating tag is placed at an image portion on a template of the commodity page. The field of "option" of the commodity database controls a new commodity list and a new commodity page. The commodities to be displayed in the new commodity list and new commodity page are limited to those identified by "N's" that are input in the field of "option", and are further limited to five or fewer commodities enumerated in counter-chronological order of "record-updating time & date." Such a commodity record-selecting condition is described in a field of "record-selecting condition" of the page-creating control file in SQL sentence, and can be changed by a user. Similarly, the field of "option" can be used for other purposes such as, e.g., a special sale or a year-end bargain sale.

[0033] In the commodity database, a field of "color & size external filename" is used to display a table of commodities defined by a plurality of standards in which different colors and sizes are available. When a template file includes a tag of "external file reference output option" ([@%file catalog.SizeColorFN@]), then a filename of a HTML tagged external file is input into the field of "color & size external filename", with the result that the content of the file is output. In this way, a table in which a variety of colors and sizes are provided for selection can easily be integrated into the commodity page.

[0034] Figs. 7 and 8 are illustrations, showing one embodiment of a page-creating control file that is made up as page-creating control information. The page-creating control file is prepared in the form of a tab-separated text file. In the file, each line generates a kind of a page. A record consists of fifteen fields. Respective titles and functions of the fields are outlined as follows:

- (1) Field of page type: type of a page to be created;
- (2) Field of title : title of a page to be created;
- (3) Field of created HTML filename: filename of a page to be created;
- (4) Field of processing condition: conditions to create a page;
- (5) Field of record-selecting condition 1: conditions written in SQL to select a record or field of a database;
- (6) Field of record-selecting condition 2: the same as the above (5), but this is a second portion divided for description therein;
- (7) Field of record-selecting condition 3: the same as the above (5), but this is a third portion divided for description therein;
- (8) Field of update-recognizing information: flag indicating whether a database as a source of information has been updated;
- (9) Field of template filename: template filename to be used;
- (10) Field of blocking factor: the maximum number of records to be displayed on a page in list pages;
- (11) Field of enumerated variable: fields of a commodity database displayed on list pages; and, (12)-(15) reserve fields.

[0035] The page-creating control file is designed to allow various page-producing conditions to be easily customized by such conditions being written to an external file as much as possible, which external file is independent of a program. Since the above-listed functions are briefly described, supplementary descriptions uncovered by the above brief description will now be mainly provided.

[0036] The "field of page type" will now be described. A page-creating program for reading in and then processing a file includes respective routines that correspond to page types. The page-creating program executes a condition decision and calculation using inherent codes. A description in the "field of page type" serves a function of indicating a code inherent to each page type. For example, referring to Fig. 3, respective page types: "categ_u" for the top page; "categ_l" for the upper classification page; "menu" for the lower classification page; "single" for the annexed pages ranging from the site guide to the home visit sales law and further displaying only shop database information; "option" for the new commodity list; and, "option_catalog" for the new commodity page. The page types are set on the basis of kinds of databases as a source of information, the number of pages to be created, and kinds of links to be created.

[0037] The "field of created HTML filename" has the same value as an HTML filename when being of a page type to generate only one page, such as "single", "categ_u", and "option." When the "field of created HTML filename" falls under any one of other page types, a filename-creating method is expressed in a simulative manner. For example, in the upper classification page (catUUnn.htm), a fixed character string of "cat" is combined with a two-digit value in a field of "upper classification code", and further with two-digit consecutive numbers (01, 02, 03, etc.), thereby creating an HTML file. In the commodity page, a value in a field of "commodity code" is a main filename, to which an extension of "htm" is added, thereby creating an HTML file. Characters improper as a filename are appropriately replaced, thereby

automatically creating an HTML file.

[0038] The "field of processing condition" defines conditions to create a page. More specifically, a predetermined description is made to the field of the processing condition on a row (record) of a related page in the page-creating control file, thereby providing control as to whether the related page is created. When the related page is not created, then the use of the under-mentioned condition output-indicating tag controls a description of link to the related page in a template file that stores page design information. Thus, a link error is prevented, which otherwise would occur when the related page is not created.

[0039] The "field of update-recognizing information" determines whether any database referenced by a created page has been updated, and then decides whether a record of the database is processed only when the database has been updated. The "field of update-recognizing information" indicates which one is referenced among flag files that corresponds to the databases. When any one of such indicated flag files has been updated after previous automatic organization, then an HTML file is created. The database and the flag file are related as follows:

Shop database:

15 shpupdttd.flg; scupdttd.flg;

commodity database:

catupdttd.flg;

upper classification table:

cuupdttd.flg; and,

20 lower classification table:

clupdttd.flg

where scupdttd.flg is updated when the field of the database referenced in the commodity page is updated.

[0040] The "field of blocking factor" refers to the number of records of the commodity database to be output to a page in list pages. Needless to say, enumerated variables as many as the blocking factor must be given in a template of the page. In addition, for example, in the upper and lower classification pages, there is an upper limit to the number of lists to be displayed on one page. Such an upper limit appears in the "field of blocking factor" of the page-creating control file. When the number of the records of the commodity database to be displayed on a page are greater than the upper limit, then a plurality of pages are created, each of which is linked to the previous and following pages through links.

[0041] The "field of enumerated variable" represents fields of the commodity database to be displayed on list pages. For example, the fields of the commodity database, which can be displayed on the upper and lower classification pages, and a combination of such fields are written in the "field of enumerated variable", and are then controlled. In the embodiments described in this specification and actual embodiments, restrictions are placed such that twenty or fewer enumerated variables are described in a template as a standard in order to permit twenty enumerated variables to be described in an HTML page. When the number of upper classifications in which commodities are present is less than twenty, then the under-mentioned link failure prevention-indicating tag inhibits a description of a link to each remaining area or each blank portion from being created, thereby preventing the occurrence of a link error. Such a restriction is to limit the number of the enumerated variables to be provided in a template, in view of an aesthetic appearance when the enumerated variables are displayed using a browser. Alternatively, other numerals may be used, depending upon a data status of the database to be actually used as well as designs of HTML pages.

[0042] In the automatic digital content-organizing system according to the present invention, the above-described structure allows a user to flexibly change specification of the system. In addition, no programmatic description is included in a template file. Instead, the template file is described using variable tags in simpler expression. Such a system of design allows for a designer's free design as well as a flexible change in the system.

[0043] Figs. 9-11 are illustrations, showing one embodiment of a template of a "top page." The template itself is an HTML file, and has variable tags inherent to the present invention embedded therein. For convenience of description, an intermediate portion of enumerated variables in the template is partially omitted.

[0044] The variable tag starts with a beginning tag that follows:

50

"[@"

[0045] The variable tag terminates with an ending tag that follows:

55

"@]"

For example, a variable name appearing in the above variable tag is provided below for a field of the shop database:

"shopinfo_ShopnameA"

For example, another variable name for the commodity database is given as follows:

5 "catalog.Code"

As seen from the above, a proper database name is preferably added to the top of the variable name in order to avoid confusing variables.

10 [0046] The link failure prevention-indicating tag will now be described. This tag may also be called a null-inhibiting tag. When a value of a variable is null (a character length is zero), then the link failure prevention-indicating tag functions to inhibit output of character strings including the previous and following descriptions with respect to the variable. An example of the link failure prevention-indicating tag is shown below:

15 [@case_not_null!@]

where "!" is a separation character. In the above example, when the following variable is null, then the previous and following character strings with respect to the following variable is inhibited from being output in order to avoid disordering a display on a screen when an image file is absent:

25 "shopinfo_ShopImage"

[0047] The condition output-indicating tag will now be described. In Fig. 11, the following tag is shown positioned at line 14:

35 [@case shopinfo_SiteType=01!
 !!@]

In the above example, when the field of "presence or absence of order page" in the shop database has a value of 01, then all descriptions in the tag are output, thereby displaying "How to Order" on the screen, and further creating a link to "order page." Meanwhile, an order page is created by a description of the page-creating control file. In this way, a site structure is controlled according to the present invention.

45 [0048] Figs. 12-16 are illustrations, showing one embodiment of a template of a "commodity page." The template contains the null-inhibiting tags) and are further written in such a manner that two or fewer images of commodities are displayed. A substitution-specifying tag is used in a field of "descriptive sentence" of the commodity database. The following is the substitution-specifying tag:

50 [@subst|catalog.Explanation!//=(br)@]

The above example indicates that two slashes "//" in the field of the descriptive sentence ("Explanation") of the commodity database is replaced by a tag of "(br)", which tag represents a line feed in HTML. The slashes "//" are substituted for data that is stored in the database at a memorandum type field by way of a line feed. This substitution is intended to avoid disordering the preceding data when the same data is output to a tab-separated text file in the course of processing. Although a sentence is not written on another line, even when a line feed code is present in HTML; the above sub-

stitution-specifying tag smoothes out such an inconvenience. Consequently, a portion in which the line feed is placed when data is entered is brought onto another line, and is then displayed on the screen.

[0049] In addition to the above, the substitution-specifying tag can be utilized to provide a display similar to that is obtained when a partitioned table is looked up. Such an example is shown below:

```

                                     [ @case not
null!sex:!catalog_Sex!!subst!01=male!02=female@]

```

In the above example, when a field of sex ("Sex") has a value of 01, then a character of "male" is displayed. When the above field has a value of 02, then a character of "female" is displayed. In addition, in the above example, a field name of "sex" is specified as null inhibition.

[0050] Further, the substitution-specifying tag can be utilized for an application in which an image file of an icon is changed according to a field of "upper classification code" of the commodity database. An example is shown below:

```
(img src="img/[ @categ.Ucode@ ].gif@ ]")
```

The above tag displays an image picture in which a value in the field of "upper classification code" is a main filename and "gif" is an extension. More specifically, the image picture consistent with the content of "upper classification code" can be displayed as an icon. Thus, the substitution-specifying tag is possible to provide various effects in simple expression. Such a tag description is similar to an HTML description, with which a Web designer is familiar. Therefore, it is easy for the Web designer to learn such a tag description.

[0051] Fig. 17 shows a "top page" displayed using a browser. The top page is created by the automatic digital content-organizing system according to the present invention. Fig. 18 illustrates a "commodity page" displayed by means of the browser. The commodity page is produced by the automatic digital content-organizing system according to the present invention. Such a refined design is difficult to acquire when the art of CGI or ASP is used. In contrast, the present invention allows for easy creation of such illustrated pages.

[0052] Although the above-described embodiments are concerned with a shopping site, the disclosed art of the present invention is not limited thereto, but is applicable to a news site, a site of public relations, a BBS site, and others. Furthermore, the present invention is applicable to automatic organization of HTML as well as other types of digital content with a hypertext link, such as XML.

[0053] As amplified in the above description, the automatic digital content-organizing system according to the present invention creates HTML pages by the steps of: indicating to a program types of the HTML pages and a method for creating the HTML pages, all of which HTML pages form a site; defining designs of the created HTML pages in combination with such indicated types and method; regularly extracting data such as commodities from a single or multiple databases; and, permitting the extracted data to be embedded in the HTML pages based on templates that include such defined designs. In the above-described system, a program element is separated from a design element. Therefore, a user and a designer are allowed to create HTML pages having enhanced designability and a Web site, and further to change designs of the HTML pages without the risk of run-away or malfunction of a program. Furthermore, the user and the designer are free to have all sites structured in accordance with their original designs. As a result, the created Web site has enhanced appeal to the viewer, thereby more successfully accomplishing an object in establishing the site. Furthermore, the user can simply enter data into the database, and then make up image data; therefore, less labor is required to add or change commodities, with consequential rapid maintenance of the Web site.

[0054] Moreover, information on a structure of a site to be created is not contained in a program, but is placed in the page-creating control means that is recorded in an external file, which external file is separate from the program. Accordingly, a template does not contain any programmatic description such as database search conditions and output indication, but is a usual HTML file having variable, tags added thereto. Therefore, for the Web designer having a profound acquaintance with HTML, it is easy to understand such a template, and further to add any description thereto. Thus, the user is allowed to flexibly change a site structure.

Claims

1. An automatic digital content-organizing system, comprising:

a means for preparing digital content including a character and an image portion by means of one of HTML and any other protocol-based digital content description language for purposes of information processing;

a database information storage means for storing one of a single database and multiple databases, said database being a source of information for creating said digital content;

a management means for administering retrieval of data from said database information storage means and input-output of the data from said database information storage means;

a page-creating control information storage means for storing page-creating control information, said page-creating control information containing a page structure and a method for creating said page structure, said page structure consisting of a plurality of kinds of pages;

a page design information storage means for storing page design information in a template that contains two tags, in which one of said tags specifies respective designs of said page structure and respective positions at which fields of said database are displayed, while another tag is a link failure prevention-indicating tag that functions to avoid including a description of a link to one of image information and any other page when said one of said image information and said other page is absent, and further to avoid including respective descriptions of the links to said image information and said other page when said image information and said other page are both absent;

a means for reading out said page-creating control information from said page-creating control information storage means;

a means for reading out information in said database from said database information storage means according to said page-creating control information, in which the information in said database is read out in a predetermined format under control of said management means;

a means for reading out optimum page design information from said page design information storage means according to said page-creating control information; and,

a program-storing means for storing a program, said program permitting such read information in said database to be embedded in said page design information, said program thereby producing pages that form said digital content.

2. An automatic digital content-organizing system as defined in claim 1, further comprising:

said page design information storage means for storing a template that contains a substitution-specifying tag, said substitution-specifying tag replacing a predetermined field of said database with a predetermined description, and then displaying such a replaced description when said database has a predetermined value in said predetermined field; and,
said program-storing means for storing a program that executes such substitutional specification.

3. An automatic digital content-organizing system as defined in claim 2, further comprising:

the page-creating control information storage means for storing said page-creating control information, to which condition output-indicating information is added, said condition output-indicating information determining on the basis of a value in a particular field of said database whether a particular page is created;
said page design information storage means for storing said page design information, to which a condition output-indicating tag is added, the condition output-indicating tag indicating depending upon a value in a particular field of said database whether a predetermined character string including a link to any other page is output; and,
said program-storing means for storing a program that executes such condition output indication.

4. A record medium having digital content recorded therein, said digital content being automatically organized by said automatic digital content-organizing system as defined in one of claims 1-3.

FIG. 1

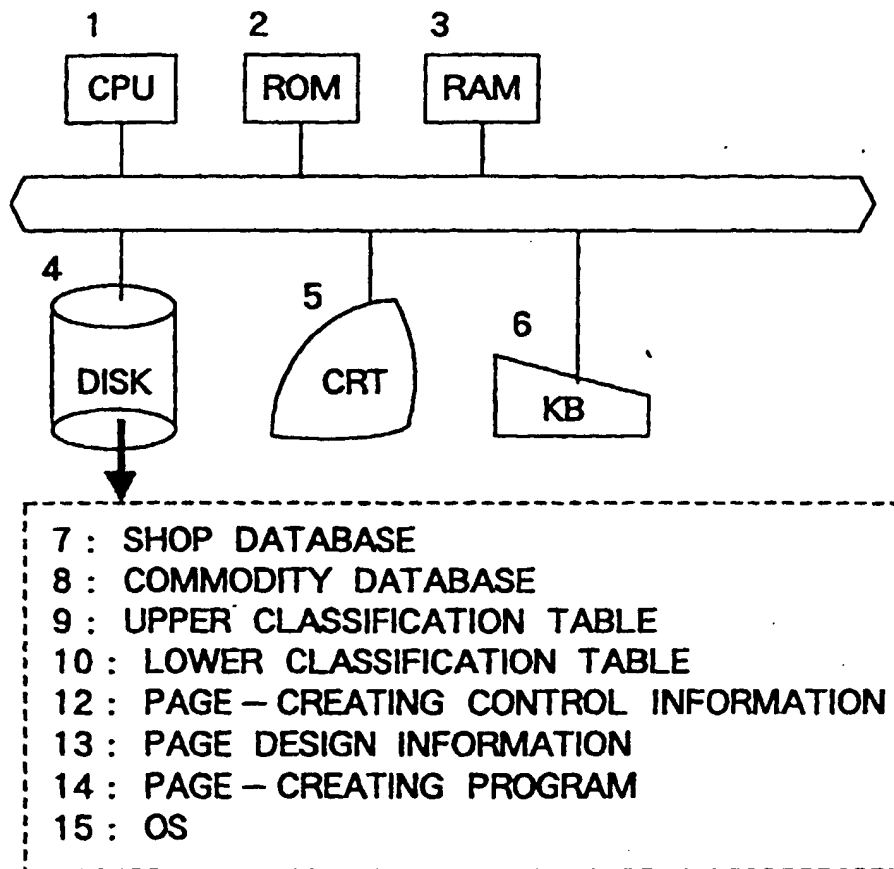


FIG. 2

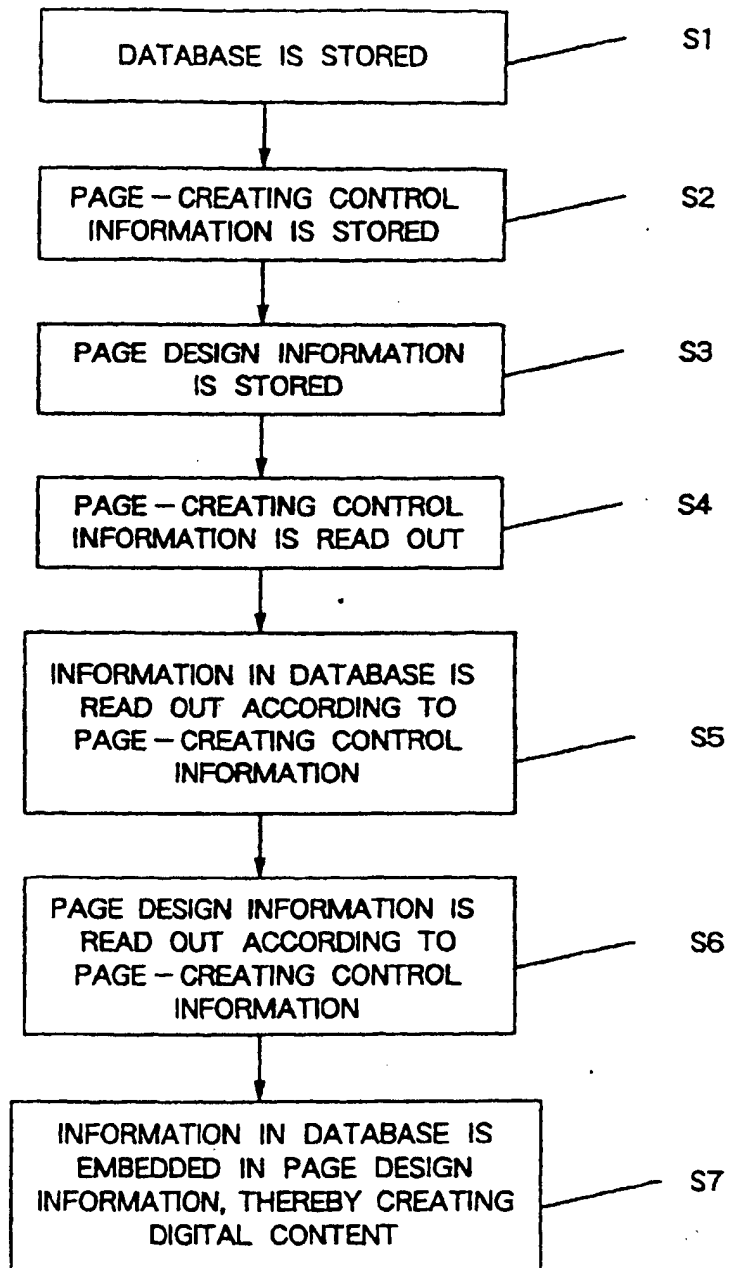


FIG. 3

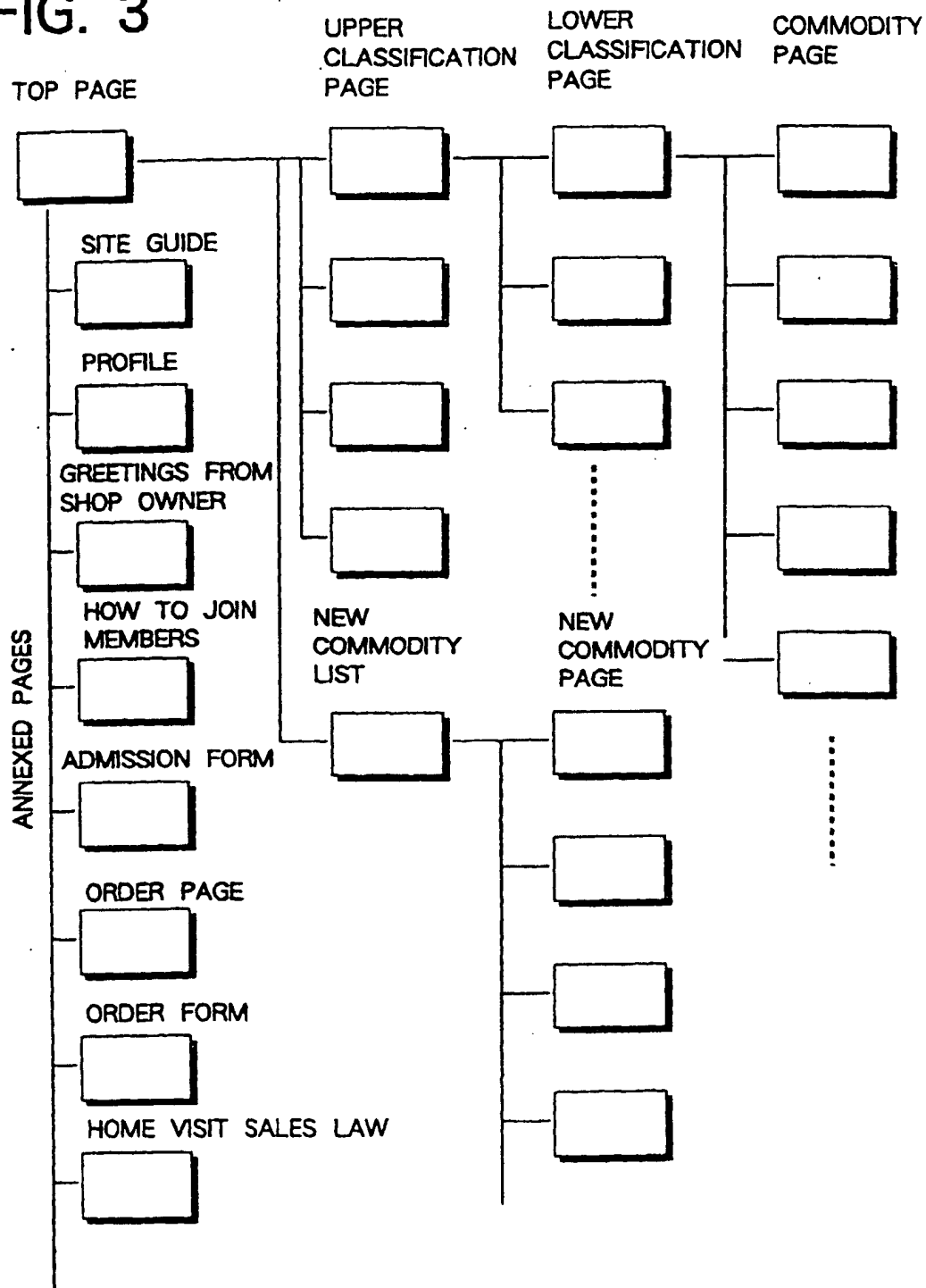


FIG.4

FIELD NAME	VARIABLE NAME	DATA TYPE	SIZE
COMMODITY CODE	Code	A	16
RECORD-ERASING TIME & DATE	RecDeleted	A	12
RECORD-CREATING TIME & DATE	RecCreated	A	12
RECORD-UPDATING TIME & DATE	RecUpdated	A	12
LOCAL PAGE-UPDATING TIME & DATE	LocalUpdated	A	12
PAGE UPDATE-RESERVING TIME & DATE	OptionUpdate	A	12
CUSTOM TEMPLATE	TemplateFN	A	12
OPTION	Option	A	10
RESERVE FIELD 1	CatReserve1	A	20
RESERVE FIELD 2	CatReserve2	A	20
HTML FILENAME	HtmlName	A	23
UPPER CLASSIFICATION CODE	ClassU	A	2
LOWER CLASSIFICATION CODE	ClassL	A	4
COMMODITY NAME	ProductName	A	30
COMMODITY NAME IN KANA-CHARACTER	ProductNameKN	A	30
COUNTRY OF PRODUCTION	MadeIn	A	40
SELLING PRICE	Price	\$	
STANDARD	Spec	A	20
COLOR	Color	A	30
COLOR & SIZE EXTERNAL FILENAME	SizeColorFN	A	64
CONSUMPTION TAX AMOUNT	Tax	\$	
PRODUCER	Maker	A	60
PRODUCER IN KANA-CHARACTER	MakerKN	A	60
PRODUCER 2	Maker2	A	60
DESCRIPTIVE SENTENCE	Explanation	M	[50]
COMMODITY IMAGE	Image	A	64
CAPTION	ImageCaption	A	40
ALTERNATIVE CHARACTER	AltCaption	A	20
COMMODITY IMAGE 2	Image2	A	64
CAPTION 2	ImageCaption2	A	40
ALTERNATIVE CHARACTER 2	AltCaption2	A	20
COMMODITY IMAGE "S"	ImageS	A	64
CAPTION "S"	CaptionS	A	40
ALTERNATIVE CHARACTER "S"	AltCaptionS	A	20

FIG.5

FIELD NAME	VARIABLE NAME	DATA TYPE	SIZE
PROCESSING OR UNPROCESSING OF MEMBERSHIP REGISTRATION	Registration	A	2
PRESENCE OR ABSENCE OF ORDER PAGE	SiteType	A	2
SHOP NAME	ShopnameA	A	100
SHOP NAME IN KANA-CHARACTER	ShopnameB	A	100
FTP HOST NAME	FtpHostname	A	64
FTP PATH	FtpPath	A	64
FTP ID	FtpID	A	20
FTP PASSWORD	FtpPassword	A	20
RESERVE 1	ShopReserve1	A	10
RESERVE 2	ShopReserve2	A	20
RESERVE 3	ShopReserve3	A	20
SHOP NAME LOGO	ShopLogo	A	64
SHOP NAME LOGO SMALL	ShopLogoS	A	64
SHOP IMAGE	ShopImage	A	64
COPY 1	Copy1	A	64
GREETINGS FROM SHOP OWNER	Message	M	[50]
SHOP OWNER'S IMAGE	ManagerImage	A	64
ADDRESS	Address	A	80
ZIP CODE	ZipCode	A	10
TELEPHONE NUMBER	Phone	A	15
FAX NUMBER	Fax	A	15
E-MAIL ADDRESS	Email	A	50
ORDER E-MAIL	OrderEmail	A	50
MEMBERSHIP REGISTRATION E-MAIL	RegistEmail	A	50
DEALER'S NAME	CompName	A	60
GENERAL MANAGEMENT EXECUTIVE	WebManager	A	30
EXTRA CHARGE OTHER THAN PRICE	ExtCharge	M	[50]
EFFECTIVE PERIOD OF ORDER	OrderTerm	M	[50]
REJECTED GOODS	Defective	M	[50]
SELLING QUANTITY	Quantity	M	[50]
DELIVERY TIME	Delivery	M	[50]
PAYMENT METHOD	PayMethod	M	[50]
COOLING-OFF PERIOD	CoolingTime	M	[50]
PARTY TO BEAR RE-POSTAGE OF RETURNED GOODS	RePostage	M	[50]
DATE OF ESTABLISHMENT	Established	A	12

FIG.6

REPRESENTATIVE	President	A	30
PROFILE	Profile	M	[50]
OTHER INFORMATION	OtherInfo	M	[50]
COPYRIGHT INFORMATION	CopyRight	M	[50]
SITE GUIDE 1	Guide1	M	[50]
SITE GUIDE 2	Guide2	M	[50]
SITE GUIDE 3	Guide3	M	[50]
SITE GUIDE 4	Guide4	M	[50]
SITE GUIDE 5	Guide5	M	[50]
TITLE OF HOW TO JOIN MEMBERS	RegistTitle	A	50
HOW TO JOIN MEMBERS	HowtoRegist	M	[50]
HOW TO ORDER	HowtOrder	M	[50]
HOW TO PAY	HowtoPay	M	[50]
NEW COMMODITY LIST TITLE	NLTitle	A	20
NEW COMMODITY LIST COPY	NLCopy	A	16
CONSUMPTION TAX CLASSIFIED CODE	Taxtype	A	2
CONSUMPTION TAX CLASSIFIED EXPRESSION	Taxtype2	A	10
TAX RATE 1	TaxRate	S	
TAX RATE 2	TaxRate2	S	

FIG.7

FIG.7A	FIG.7C
FIG.7B	FIG.7D

FIG.7A

PAGE TYPE	TITLE	CREATED HTML	PROCESSING CONDITION	RECORD-SELECTING CONDITION 1	RECORD-SELECTING CONDITION 2
common_data	COMMON VARIABLE, BASIC SHOP INFORMATION			SELECT * FROM 'shopinfo.db'	
categ_u	TOP PAGE	local\index.htm	always	SELECT DISTINCT D.* FROM 'categ_u.db' D, 'categ_l.db' D1, 'catalog.db' D2	WHERE (D2.RecDeleted = " ") AND (D2.ClassU = D.ClassU_code) AND (D2.ClassL = D1.ClassL_code)
single	SITE GUIDE	local\guide.htm	always		
single	PROFILE	local\profile.htm	always		
single	GREETINGS FROM SHOP OWNER	local\master.htm	always		
single	HOW TO JOIN MEMBERS	local\regist.htm	case shopinfo_Registr ation=01		

FIG.7B

single	ADMISSION FORM	local\regform.htm	case shopinfo_Registr ation=01		
single	ORDER PAGE	local\order.htm	case shopinfo_SiteTyp e=01		
single	HOME VISIT SALES LAW PAGE	local\law.htm	case shopinfo_SiteTyp e=01		
single	ORDER FORM	local\orform.htm	case shopinfo_SiteTyp e=01		
single	FAX PAPER SHEET	local\faxform.htm	case shopinfo_SiteTyp e=01		

FIG.7C

RECORD-SELECTING CONDITION 3	UPDATE-RECOGNIZING INFORMATION	TEMPLATE FILE	BLOCKING FACTOR	ENUMERATED VARIABLES
ORDER BY D.ClassU_code	shpupdttd.flg	template\index.htm	20	ClassU_code!ClassU_desc! ClassU_Comment! ClassU_Image
	shpupdttd.flg	template\guide.htm		
	shpupdttd.flg	template\profile.htm		
	shpupdttd.flg	template\master.htm		
	shpupdttd.flg	template\regist.htm		

FIG. 7D

	shpupdttd.flg	template\regfor m.htm		
	shpupdttd.flg	template\order.h tm		
	shpupdttd.flg	template\law.htm		
	shpupdttd.flg	template\orform. htm		
	shpupdttd.flg	template\faxform .htm		

FIG.8

FIG.8A	FIG.8C
FIG.8B	FIG.8D

FIG.8

FIG.8A	FIG.8C
FIG.8B	FIG.8D

FIG.8A

PAGE TYPE	TITLE	CREATED HTML	PROCESSING CONDITION	RECORD-SELECTING CONDITION 1	RECORD-SELECTING CONDITION 2
categ_l	UPPER CLASSIFICATION PAGE	local\categUUnn.htm	always	SELECT DISTINCT D.*, D1.* FROM 'categ_u.db' D, 'categ_l.db' D1, 'catalog.db' D2	WHERE (D2.RecDeleted = "") AND (D2.ClassU = D.ClassU_code) AND (D2.ClassL = D1.ClassL_code)
menu	LOWER CLASSIFICATION PAGE	local\UUULLLLnn.htm	always	SELECT DISTINCT D.*, D1.*, D2.Code, D2.HtmlName, D2.ProductName, D2.Price, D2.Images FROM 'categ_u.db' D, 'categ_l.db' D1, 'catalog.db' D2	WHERE (D2.RecDeleted = "") AND (D2.ClassU = D.ClassU_code) AND (D2.ClassL = D1.ClassL_code)
catalog	COMMODITY PAGE	local\goods\CCCCC\CC.htm	always	SELECT D.*, D1.ClassU_desc, D2.ClassL_desc From 'catalog.db' D, 'categ_u.db' D1, 'categ_l.db' D2	WHERE (D.RecDeleted = "") AND (D.ClassU = D1.ClassU_code) AND (D.ClassL = D2.ClassL_code)

FIG.8B

option	NEW COMMODITY LIST	local%new.htm	always	SELECT DISTINCT D.*, D1.*, D2.Code, D2.HtmlName, D2.ProductName, D2.Price, D2.Image, D2.RecUpdated FROM 'categ_u.db' D, 'categ_l.db' D1, 'catalog.db' D2 WHERE (D2.RecDeleted = "") AND (D2.'Option' LIKE '%N%') AND (D2.ClassU = D.ClassU_code) AND (D2.ClassL = D1.ClassL_code)
option_catalo g	NEW COMMODITY PAGE	local%newnnn.htm	always	WHERE (D.RecDel ted = "") AND (D.'Option' LIKE '%N%') AND (D.ClassU = D1.ClassU_code) AND (D.ClassL = D2.ClassL_code)

FIG.8C

RECORD-SELECTING CONDITION 3	UPDATE- RECOGNIZING INFORMATION	TEMPLATE FILE	BLOCKING FACTOR	ENUMERATED VARIABLES
ORDER BY D.ClassU_code, D1.ClassL_code	shpupdt.d.fig:cuu pdt.d.fig:clupdt.d.fig g	template%catUU nn.htm	20	ClassL_code:Class L_desc:ClassL_Com ment:ClassL_Image
ORDER BY D.ClassU_code, D1.ClassL_code	shpupdt.d.fig:cuu pdt.d.fig:clupdt.d.fig g:catupdt.d.fig	template%UUULLL Lnn.htm	20	catalog.Code:catal og.HtmlName:catal og.ProductName:c atalog.Price:catalo g.Images
ORDER BY D.RecUpdated DESC	scupdt.d.fig:cuup dt.d.fig:clupdt.d.fig !catupdt.d.fig	template%CCCC CCCC.htm		

FIG.8D

ORDER BY D2.RecUpdated DESC, D.ClassU_code, D1.ClassL_code	shpupdttd.fig/catu pdttd.fig	template\new.ht m	5	catalog_Code:catal og_ProductName:c atalog_Price:catalo g_Image
ORDER BY D.RecUpdated DESC	shpupdttd.fig/catu pdttd.fig	template\newnn. htm	5	

FIG.9

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="ja">
<head>
<title>[@shopinfo_ShopnameB@]</title>
<META name="author" content="[@shopinfo_ShopnameA@]">
<META name="description" content="[@shopinfo_Copy1@]">
</head>
<body bgcolor="#ffffff" text="#000000" link="#660000" vlink="#000066" alink="#cccccc">
<center>
<!--*****TILE-->
[@case_not_null!@]
<p>
[@case_not_null!@]
</p>
<p>
<font size=5 color="#660000"><strong>[@substshopinfo_Copy1!/=<br>@]</strong></font>
</p>
<!--*****MENU-->
<table border="0" width="465">
<tr>
<td align="left" valign="top" width="50"></td>
<td align="left" valign="middle">
[@case_not_null!<a href="cat!ClassU_code01!01.htm">@]<font size="3"><strong>[!@ClassU_
desc01@]</strong></font></a>

```

FIG.10

```

[ @case_not_null||| <a href="cat!ClassU_code02!01.htm">0]<font size="3"><strong>[ @CclassU
_desc020]</strong></font></a>
[ @case_not_null||| <a href="cat!ClassU_code03!01.htm">0]<font size="3"><strong>[ @CclassU
_desc030]</strong></font></a>
[ @case_not_null||| <a href="cat!ClassU_code04!01.htm">0]<font size="3"><strong>[ @CclassU
_desc040]</strong></font></a>
..... 18 lines are omitted .....
[ @case_not_null||| <a href="cat!ClassU_code19!01.htm">0]<font size="3"><strong>[ @CclassU
_desc190]</strong></font></a>
[ @case_not_null||| <a href="cat!ClassU_code20!01.htm">0]<font size="3"><strong>[ @CclassU
_desc200]</strong></font></a>
<br></td>
</tr>
<tr>
<td align="left" valign="top" width="50"></td>
<td align="left" valign="middle"><a href="new.htm"><font size="4"><strong>[ @shopinfo_
_NLCopy0]</strong></font></a></td></tr></table>
<br>
<!--*****SHOP INFORMATION-->
<table border="0">
<tr>
<td align="left" valign="top" colspan="2" width="2" width="462"><br></td>
</tr>
<tr>

```

FIG.11

```

<td align="left" valign="top" width="180" nowrap><a href="guide.htm"></a><a href="profile.htm"></a><a
href="master.htm"></a></td>
<td align="left" valign="top"><font size=2 color="#666666">〒[0shopinfo_ZipCode0] [0
shopinfo_Address0]<br>TEL:[0shopinfo_Phone0] FAX:[0shopinfo_Fax0]<br>[0case_not_nul1]<
a href="mailto:!shopinfo_Email1">[0shopinfo_Email0]</a>[0]</td>
</tr>
<tr>
<td align="left" valign="top" colspan="2" width="460"><br><font size=2 color="#666666">[0shopinfo_CopyRight0]</font><br></td>
</tr>
<tr>
<td align="right" valign="top" colspan="2" width="460">[0case shopinfo_SiteType=01]<a
href="order.htm"></a>[0][0case shopinfo
Registration=01]<a href="regist.htm"></a>[0]</td>
</tr>
</table>
<!--*****-->
</center>
</body>
</html>

```

FIG.12

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="ja">
<head>
<title>[catalog_ProductName0]</title>
</head>
<body bgcolor="#ffffff" text="#000000" link="#660000" vlink="#660000" alink="#cccccc">

<center>

<!--*****#NAVIGATION-->
<table border="0" cellpadding="0">
<tr>
<td align="left" valign="bottom" nowrap>[@case_not_null!@]</td>
<td align="right" valign="bottom" nowrap><a href="..index.htm"></a></td>
</tr>
<tr>
<td align="left" valign="top" colspan="2"></td>
</tr>
<tr>
<td align="right" valign="top" colspan="2"><font size="2"> | [0case_not_null!<a href=".. /
cat[catalog_ClassU!01.htm">[0ClassU_desc0] list</a>@] | [0case_not_null!<a href=".. / [0catalog
_ClassU0]!catalog_ClassL!01.htm">[0ClassL_desc0] list</a>@] | <font></td>
</tr>
</table>

```


FIG.13

```

<br>
<!--*****-->
<table border="0" cellpadding="0" width="550" cellspacing="0">
<tr>
<td align="right" valign="top" width="550" colspan="3"><strong><font size=5>[@catalog_
ProductName@]</font></strong><br><hr><
/td>
</tr>
<tr>
<td align="left" valign="top" width="550" colspan="3">
<table border="0" cellpadding="0" width="550" cellspacing="0">
<tr>
<td align="right" valign="top" width="18"></td>
<td align="left" valign="top" width="532"><strong><font size=2>[@catalog_Code@] <
/ font><font size=3>[@catalog_Maker@][@case_not_null!catalog_MakerKN!@] </font>[@case_not
_null!<font size=2>!catalog_Maker2!</font>@]<font size=4>[@catalog_ProductName@]</font></
strong>[@case_not_null!<font size=2>!catalog_ProductNameKN!</font>@][@case_not_null!<font
size=4>COUNTRY OF PRODUCTION (!catalog_MadeIn!) </font>@]<br></td>
</tr>
</table>
</td>
</tr>
<tr>
<td align="center" valign="top" width="200"><br>

```

FIG.14

```

[@case_not_null!@<br>[@
case_not_null!<font size="2"><strong>!catalog_ImageCaption!</strong></font><@<br>

[@case_not_null!@<br>[@
case_not_null!<font size="2"><strong>!catalog_ImageCaption2!</strong></font><@</td>

<td align="left" valign="top" width="10"></td>
<td align="left" valign="top" width="340">    <br><br><font size=3>¥[@
catalog_Price@][@case shopinfo_Taxtype=01!(<+CONSUMPTION TAX¥!catalog_Tax!)<@][@case
shopinfo_Taxtype=00!(<!shopinfo_Taxtype2!)<@</font><br><br>
<table border="0" cellpadding="2">
[@case_not_null!<tr><td align="left" valign="top"><strong><font size=2>COLOR</font></strong>
<br></td><td align="middle" valign="top"><font size=2>!catalog_Color!</font></td></tr></td>
[@case_not_null!<tr><td align="left" valign="top"><font size=2><strong>STANDARD</strong></
font><br></td><td align="middle" valign="top"><font size=2>!catalog_Spec!</font></td></tr>
</td>
[@%file catalog_SizeColorFN@]
</td>
</table>
[@subst!catalog_Explnation!//=<br><@]
<br><br></td>
</tr>
</table>

```

FIG.15

```

<br>
<!--*****NAVIGATION-->
<table border="0" cellpadding="0">
<tr>
<td align="left" valign="bottom"><font size="2"><strong>[0case shopinfo_SiteType=01!
<a href="../orform.htm" target="_blank">●ORDER FORM</a> <a href="../faxform.htm" target="
_blank">●FAX PAPER SHEET</a> <a href="../order.htm">●HOW TO ORDER</a>!!0] [0case
shopinfo_Registration=01!<a href="../regist.htm">●HOW TO JOIN MEMBERS</a></strong></font>
</td>!!0]
<td align="right" valign="bottom"><a href="../index.htm"></a></td>
</tr>
<tr>
<td align="left" valign="top" colspan="2"></td>
</tr>
<tr>
<td align="right" valign="top" colspan="2"><font size="2">| <a href="../cat[0catalog_ClassU
0]01.htm">[0ClassU_desc0] list | <a href="../[0catalog_ClassU0] [0catalog_ClassU0]01.htm">[0
ClassL_desc0] list<font></td>
</tr>
</table>
<br>
<!--*****SHOP ADDRESS ETC. -->
<table border="0" width="580" cellpadding="0">

```

FIG.16

```

<tr>
<td align="left" valign="top" width="580"><font size="2" color="#000000">〒[@shopinfo_
ZipCode@][@shopinfo_Address@]<br>TEL :[@shopinfo_Phone@] FAX :[@shopinfo_Fax@]  [@case_not
_null!]<a href="mailto:shopinfo_Email!">[@shopinfo_Email@]</a></font><@></td>
</tr>
</table>

<!--*****-->
</center>

</body>
</html>

```

FIG.18

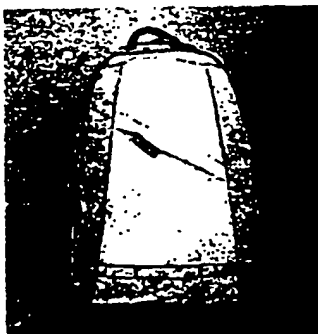
Green Field

QooTOP

| EAST MOUNTAIN LIST | ZACK LIST |

PORTERS DAYPACK

▲ 1001 EAST MOUNTAIN PORTERS DAYPACK
COUNTRY OF PRODUCTION(JAPAN)



GOOD QUALITY

¥ 14500(+CONSUMER TAX ¥ 725)

COLOR PINK, BLUE

SPEC 25LITTER

DETAILED EXPLANATION OF THIS
COMMODITY.....



BLUE COLOR VERSION

● ORDER FORM ● HOW TO ORDER ● HOW TO JOIN MEMBERS

QooTOP

| EAST MOUNTAIN LIST | ZACK LIST |

ZIP CODE 144-0051 6-36-11, NISHIKAMATA OHTA-KU TOKYO
PHONE:03-53XX-4628 FAX:03-53XX-4629 MAIL@ADDRESS